

HY_TEX User's Manual

Rob J Hyndman

Version 7.0, April 2007

HY_TEX is a L^AT_EX style file for producing even better documents. It is a collection of commands that I have found useful and think other L^AT_EX users probably will too. It is intended for use with L^AT_EX 2_ε.

HY_TEX 7.0 is a trimmed down version of HY_TEX 6.0. As more packages have become available, some of the facilities in HY_TEX became obsolete. In particular:

- For compact lists, use the `paralist` package.
- For variations on description lists, use the `mdwlist` package.
- For bibliographies, use Bib_TEX with the `natbib` package.
- For bold mathematical symbols, use the `bm` package.
- For blackboard letters, use the `amsfonts` package.
- For variations on the `verbatim` environment, use the `verbatim` package.
- For better control in typesetting theorems, use the `ntheorem` package.
- For better control over numbering equations and floats, use the `amsmath` package. (This is pretty useful for lots of other things too.)

Other trimming occurred by removing facilities that were hardly ever used.

To use HY_TEX, simply begin your L^AT_EX document like this:

```
\documentclass{article}
\usepackage{hytex}

\begin{document}
```

There is often no need for any other preamble. The page dimension commands that many people include in the preamble are unnecessary as HY_TEX does all this for you. HY_TEX may also be used with the `report`, `book` or `letter` document styles.

When the HY_TEX style file is loaded in this way, a number of additional commands become available. Almost everything else in L^AT_EX will work exactly as normal.

It is assumed in this manual, that readers are familiar with L^AT_EX as described in Kopka and Daly's excellent book "Guide to L^AT_EX".

If you have any suggestions for inclusion in future versions of HY_TEX, or if you have found a bug, please let me know at `Rob.Hyndman@buseco.monash.edu.au`.

Some of the HY_TEX commands I have borrowed from similar style files developed by other L^AT_EXers.

Abbreviations Some LaTeX commands are typed repeatedly. Therefore, I have provided some abbreviations for commonly used sequences of commands. Of course, the full commands can still be used. The abbreviations are:

```
\bt    \begin{center}\begin{tabular}
\et    \end{tabular}\end{center}
\biz    \begin{itemize}
\eiz    \end{itemize}
\ben    \begin{enumerate}
\een    \end{enumerate}
```

Blank page The command `\blankpage` will give you just that!

Boxed equations An equation can be placed in a box using the command `\eqnbox`:

```
$$\eqnbox{\E\hat f (y|x) - f(y|x) = \sigma_K^2
\left\{ a^2 \frac{h'_X(x)}{h_X(x)} \frac{\partial f(y|x)}{\partial x} + \frac{1}{2} a^2 \frac{\partial^2 f(y|x)}{\partial x^2} + \frac{1}{2} b^2 \frac{\partial^2 f(y|x)}{\partial y^2} \right\}}
\smallfrac{1}{2} a^2 \frac{\partial^2 f(y|x)}{\partial x^2} +
\smallfrac{1}{2} b^2 \frac{\partial^2 f(y|x)}{\partial y^2} \right\}$$
```

produces

$$E\hat{f}(y|x) - f(y|x) = \sigma_K^2 \left\{ a^2 \frac{h'_X(x)}{h_X(x)} \frac{\partial f(y|x)}{\partial x} + \frac{1}{2} a^2 \frac{\partial^2 f(y|x)}{\partial x^2} + \frac{1}{2} b^2 \frac{\partial^2 f(y|x)}{\partial y^2} \right\}$$

Boxed paragraphs To place a paragraph (including displayed equations) in a centered box, use the command `\boxpar`. For example,

```
\boxpar{10.5cm}{\textbf{Definition:} If $Y_1, \dots, Y_n$ denote observations
with mean $\bar{Y} = \frac{1}{n} \sum_{i=1}^n Y_i$, then their \textit{sample
standard deviation} is defined as
$$s = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (Y_i - \bar{Y})^2}.$}$}
```

produces

Definition: If Y_1, \dots, Y_n denote observations with mean $\bar{Y} = \frac{1}{n} \sum_{i=1}^n Y_i$, then their *sample standard deviation* is defined as

$$s = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (Y_i - \bar{Y})^2}.$$

Note that you must specify the width of the box.

If the `color` package is loaded, you can also `\shadexbox` in the same way, but the box will be shaded.

Chapter heads Even among well printed books, there is great variation in the format of section heads. The font may be bold, sans serif, or small caps, in sizes ranging from normal to rather large. If a section number is present, it may be followed by a period or by a long space. The changes here give the casual LaTeXer more control over the style of headings.

The format of chapter heads may be controlled by the command

```
\mychapter{<head1>}{<head2>}
```

Here <head1> is the chapter heading to use with the `\chapter` command (when the chapter is numbered) and <head2> is the heading to use with the `\chapter*` command (when the chapter is not numbered). The text of the `\chapter` command is referenced in <head1> and <head2> by writing #1.

For example, the default chapter head is defined by

```
\mychapter{\huge\bf\chapapp\thechapter\[\[20pt]\Huge\bf #1}
{\Huge \bf #1}
```

Here, the command `\chapapp` is usually defined to be “Chapter” but the L^AT_EX command `\appendix` changes it to “Appendix”. The command `\thechapter` refers to the chapter number.

If you prefer smaller headings in smallcaps without the word “Chapter” prefixed, you could try:

```
\mychapter{\LARGE \sc \thechapter. #1}{\LARGE \sc #1}
```

There are three inbuilt chapter formats.

<code>\sfchapter</code>	Makes chapter headings in sans serif font
<code>\sfbchapter</code>	Makes chapter headings in sans serif bold font
<code>\boxchapter</code>	Places chapter number in a box

Any of these commands can be placed in the preamble.

Chapters always begin on a new page; `\chaptopsep` is the amount of blank space at the top of the page before the chapter head, and `\chapaftersep` is the amount of blank space placed between the chapter head and the text. They are initially 1.8cm and 1.4cm respectively. They may be changed by the `\setlength` command.

An alternative approach is possible using the `fncychap` package.

Columns Use the package `multicol` to set text in columns. H_MT_EX adds a new command `\newcolumn` to force a column break.

Date and Time To get today's date, use the T_EX command `\today`. H_MT_EX adds an optional argument to this command to control the date format and to provide Australian rather than American formats. The options are given below with examples of the format.

0 (default)	23 July 1999
1	23 Jul 99
2	Friday, 23 July 1999
3	23.07.99

Other formats can be created using the commands `\Dow`, `\Date`, `\Month`, `\Mon`, `\Year` and `\Yr`.

The current time is available using the new command `\clock`. For example, if we type

This document was processed at `\clock\` on `\today`.

the printed document will read

This document was processed at 8.18pm on 3 April 2007.

This is used in the command `\draft` to add a time stamp to each page.

These commands refer to the date and time at which the dvi file is produced.

Double spacing For theses and journal articles, it is often necessary to use widely spaced text. This is made possible using the `setspace` package. The command `\doublespaced` in the preamble will give ‘double-spaced’ text throughout the document. Single spacing can be turned back on for parts of the document (such as for tabular) using `\begin{singlespace}` and `\end{singlespace}`. Note that double spacing is automatically turned off within footnotes and floats (figures and tables).

The amount of spacing can be controlled via the `\setspacing` command. The command `\doublespaced` actually uses `\setspacing{1.75}`. If this is too much, try `\setspacing{1.5}` (or some appropriate value) in the preamble. `\setspacing` can be used throughout the document to change spacing.

The `newspacing` environment can be used for setting the spacing within a section. e.g., `\begin{newspacing}{1.6} ... \end{newspacing}`.

Drafts Placing the command `\draft` in the preamble has the effect of printing DRAFT followed by the time and date at the bottom of each page. This is useful for keeping track of documents for which many drafts are printed. The `color` package must be loaded first.

Figures To include graphics, put the command `\usepackage{graphicx}` in the preamble. The graphs can be pulled in with the command

```
\graph[<placement>]{<options>}{<file>}{<caption>}
```

The first argument is optional and positions the figure (as with the `table` or `figure` environment). The second argument provides any optional arguments to the `\includegraphics` command; it can be left empty but would normally contain something like `width=15cm`. The third argument is the name of the file. Note the file extension is not required—it is assumed to be `eps` if `LaTeX` is used and `jpg` or `pdf` if `pdfLaTeX` is used. The caption is given in the fourth argument.

All figures are automatically numbered (using the figure counter). They may be referenced by their file name. For example `Figure \ref{file}` will produce the text “Figure 3” if it is the third figure to be included in the document.

Fonts **Bold sans serif fonts** can be obtained using `\sfb` instead of `\sf` or `\textsfb` instead of `\textsf`.

PostScript fonts are available using, for example, `\timesroman` in the preamble. This changes the roman font to Times-Roman, but leaves the maths, sans serif and typewriter fonts unchanged. This is not the same as using the `times` package which changes the roman, sans serif and typewriter fonts. Similarly,

\bookman , \palatino , \newcentury , \avantgarde , \helvetica , \courier change to other PostScript fonts. Several of these commands can be used together to define san serif, roman and typewriter fonts.

To have both maths and text in palatino font, use the package \mathpazo .

To obtain page numbers in sans serif, place \sfpage in the preamble. For sans serif section headings use \sfsection , and for sans serif chapter headings use \sfchapter . Also, \sfbpage , \sfbsection and \sfbchapter work as above. This document uses \sfbsection and \sfbpage .

Headers and footers The default pagestyle produces a ‘running header’ containing section names and page numbers. The headers and footers can be easily modified using the following commands.

The commands

```

 $\text{\lhead}\{<item>\}$    $\text{\chead}\{<item>\}$    $\text{\rhead}\{<item>\}$ 
    
```

set the left, center, and right parts of the headers. The corresponding commands for footers are

```

 $\text{\lfoot}\{<item>\}$    $\text{\cfoot}\{<item>\}$    $\text{\rfoot}\{<item>\}$ 
    
```

The first version of these commands was developed by Lance Berc. They have been modified for use in \HTeX .

If the \twoside style option is invoked before \HTeX , the macros switch the right and left items on even numbered pages. If you require different headers and footers for even numbered pages, as often with books, use the commands

```

 $\text{\elhead}\{<item>\}$    $\text{\echad}\{<item>\}$    $\text{\erhead}\{<item>\}$ 
 $\text{\elfoot}\{<item>\}$    $\text{\ecfoot}\{<item>\}$    $\text{\erfoot}\{<item>\}$ 
    
```

to define them. They should follow the commands, \lhead , \chead , etc.

These commands work with the pagestyle $\text{\threepartheadings}$ which is the default. If no headers and footers are desired, use the command $\text{\pagestyle}\{empty\}$ in the preamble.

To force page headers to be uppercase, use the command \uppercasehead .

The preamble command \underhead places a rule of length \textwidth and width \underheadwidth a distance of \underheadsep under the header text. The preamble command \boxhead places a box around the header as shown in this document. Shaded headers are available using \shadehead although the \graphicx package must be loaded before using this command and it is only available for postscript printers. For footers use \overfoot , \boxfoot and \shadefoot .

The font used in headers can be changed using \headstyle . For example, to make all headers sans serif, put $\text{\headstyle}\{sf\}$ in the preamble (as was done for this document). For footers, use \footstyle . Any font command included in a \lhead , \rhead , etc., will override the \headstyle . The default is $\text{\headstyle}\{sl\}$.

Marginal notes `\mnote{Comment}` can be used to place text in the margins. `\setmarginparleft` or `\setmarginparright` should be called first to ensure there is enough space in the margins. The text is set in `footnotesize`.

Mathematical symbols Several new commands have been added to produce mathematical symbols:

<code>\invstackrel{}{}{}</code>	Same as <code>\stackrel</code> but inverted. E.g., $\mathop{\longrightarrow}\limits_{n \rightarrow 0}$ produces $\mathop{\longrightarrow}\limits_{n \rightarrow 0}$
<code>\dist</code>	Produces the ‘distributed as’ symbol, $\stackrel{d}{=}$
<code>\smallfrac{}{}{}</code>	Produces small fraction like $\frac{1}{2}$
<code>\bddots</code>	Produces backwards diagonal dots: $\cdot\cdot$

The E symbol for expectation and Pr for probability should appear in roman font rather than maths italics when used in equations. This can be easily achieved using `\E` and `\Pr`. Similarly, `\var`, `\cov` and `\corr` produce Var, Cov and Corr in maths mode. To obtain these in sans serif font use `\sfE` somewhere before you need them.

Page size The command `\a4page` makes the page size A4. A similar command `\letterpage` does the same for the US letter size. A4 is the default.

Three new commands assist with setting the page size. `\setsidemargin` describes the space to appear in the margins on each side of the page. These are equal in size by default. The command `\setbinding` adds some space to the inner margin in case the document is to be bound. The default settings have zero binding space and 3cm side margins. Note these commands look after the settings for `\oddsidemargin` and `\evensidemargin` and take into account whether the page is A4 or US letter.

The command `\setmarginparleft` is used to set a wide margin on the left (or odd side) of the page. The argument is the length of `\marginparwidth`. It also ensures all marginal notes are on the inner side of the page

The command `\setmarginparright` does the same but on the right (or even side) of the page.

Paragraph indentation By default, \LaTeX sets paragraph indentation to zero and the default inter-paragraph space is 1.8 ex.

Customised changes are possible using `\setlength`.

Quotations There is a new environment `smallquote` which is identical to `quote` except that the quotation is set in a small font.

Section heads The format of section heads may be controlled by the command

$$\text{\mysection{<sec>}{<style>}{<preface>}}$$

Here `<sec>` is the section level: `section`, `subsection`, etc. The style in which the section is to be set is specified by `<style>`; e.g., `\large\sf`. The argument

<preface> is the text to be set before the text of the section head — usually \thesection and some punctuation. The defaults are as follows.

```
\mysection{section}{\Large\bf}{\thesection~~}
\mysection{subsection}{\large\bf}{\thesubsection~~}
\mysection{subsubsection}{\normalsize\bf}{\thesubsubsection~~}
\mysection{paragraph}{\normalsize\bf}{\theparagraph~}
\mysection{subparagraph}{\normalsize\bf}{\thesubparagraph~}
```

To obtain section headings in a sans serif font, include \sfsection in the preamble. For bold sans serif font, put \sfbsection in the preamble.

To obtain headings in the left margin, use the command \setmarginparleft in the preamble, then use \msection in place of \section and \msubsection in place of \subsection. This is what was done for this document.

The space before and after section headings can be altered by changing the values of \presectionsip and \postsectionsip. Similarly for subsections. But note that the space after the section headings must be positive (I don't know why).

For an alternative approach, use the sectsty package.

Table of contents In technical articles it is sometimes convenient to place a table of contents on the first page, right after the title and abstract. The section entries in the default table of contents for the article style tend to be too widely spaced for this purpose. The command \tighttoc produces a tighter table of contents. It was used to generate the table of contents on the first page.

The command \tighttoccols will produce a table of contents in two columns. Note that you need to use the package multicol for this to work.

Tables Often, you may wish to produce a column of figures aligned on the decimal point while using tabular. The package dcolumn will provide this facility.

You can set all lines in a table to a certain thickness using a standard L^AT_EX command. For example, \setlength{arrayrulewidth}{0.5mm} sets all lines to have width 0.5mm. Where you want some lines thicker than others, HY_TE_X provides two new commands, \bhline and \bvline to be used in place of \hline and \vline. The next example illustrates the use of \bhline; the use of \bvline is similar.

<pre>\begin{tabular}{c} \bhline{4} \bf Numbers\\ \hline 123\\ 456\\ 789\\ \bhline{4} \end{tabular}</pre>	which produces	<table> <tr> <th>Numbers</th> </tr> <tr> <td>123</td> </tr> <tr> <td>456</td> </tr> <tr> <td>789</td> </tr> </table>	Numbers	123	456	789
Numbers						
123						
456						
789						

The thicker lines are four times the width of the other lines in this case. Different

multiples are obtained with different arguments.

Tabs A new command `\tab` will insert some hard space providing the ability to inset text using tabbing, rather like a word processor. The command takes a numerical argument which gives the distance to tab in as a multiple of `\tabwidth`. The default value of `\tabwidth` is 1cm. So, for example, `\tab3` will tab in 3cm.

Titles The command `\hytextitle` can be used in place of `\maketitle` for the documentstyle 'article'. It takes the arguments from `\title`, `\author` and `\date` but formats them differently from `\maketitle`. There is also a `\boxtitle` command which, if used before `\hytextitle`, causes the title to appear in a box. The front page of this document gives an example of the use of `\boxtitle` with `\hytextitle`.

The environment `hytexboxtitle` does a similar thing, except all text in the environment appears below the title. This is particularly useful for abstracts. The environment `hytextitlepage` is the same as `hytexboxtitle` except it causes subsequent material to appear on a new page.

To make the title appear in the header, put `\titlehead` in the preamble. The font used for the title and author can be altered using `\titlefont` and `\authorfont`. (The date is in the same font as the author.)

Two up To obtain pages "two up" (i.e., two pages printed at half size on each sheet of paper), use the command `\twoup` in the preamble. This calls the package `2up` and figures out the page dimensions for you.